In [1]:

```
!pip install yfinance==0.1.67
!pip install pandas==1.3.3
!pip install requests==2.26.0
!pip install plotly==5.3.1
```

```
Requirement already satisfied: yfinance==0.1.67 in c:\programdata\anaconda3\lib
\site-packages (0.1.67)
Requirement already satisfied: pandas>=0.24 in c:\programdata\anaconda3\lib\sit
e-packages (from yfinance==0.1.67) (1.3.3)
Requirement already satisfied: requests>=2.20 in c:\programdata\anaconda3\lib\s
ite-packages (from yfinance==0.1.67) (2.26.0)
Requirement already satisfied: numpy>=1.15 in c:\programdata\anaconda3\lib\site
-packages (from yfinance==0.1.67) (1.21.5)
Requirement already satisfied: multitasking>=0.0.7 in c:\programdata\anaconda3
\lib\site-packages (from yfinance==0.1.67) (0.0.11)
Requirement already satisfied: lxml>=4.5.1 in c:\programdata\anaconda3\lib\site
-packages (from yfinance==0.1.67) (4.9.1)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\programdata\anacond
a3\lib\site-packages (from pandas>=0.24->yfinance==0.1.67) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in c:\programdata\anaconda3\lib\sit
e-packages (from pandas>=0.24->yfinance==0.1.67) (2022.1)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\sit
e-packages (from requests>=2.20->yfinance==0.1.67) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\l
ib\site-packages (from requests>=2.20->yfinance==0.1.67) (2022.9.14)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\programdata\anaconda
3\lib\site-packages (from requests>=2.20->yfinance==0.1.67) (1.26.11)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\programdata\anac
onda3\lib\site-packages (from requests>=2.20->yfinance==0.1.67) (2.0.4)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-pa
ckages (from python-dateutil>=2.7.3->pandas>=0.24->yfinance==0.1.67) (1.16.0)
Requirement already satisfied: pandas==1.3.3 in c:\programdata\anaconda3\lib\si
te-packages (1.3.3)
Requirement already satisfied: pytz>=2017.3 in c:\programdata\anaconda3\lib\sit
e-packages (from pandas==1.3.3) (2022.1)
Requirement already satisfied: numpy>=1.17.3 in c:\programdata\anaconda3\lib\si
te-packages (from pandas==1.3.3) (1.21.5)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\programdata\anacond
a3\lib\site-packages (from pandas==1.3.3) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-pa
ckages (from python-dateutil>=2.7.3->pandas==1.3.3) (1.16.0)
Requirement already satisfied: requests==2.26.0 in c:\programdata\anaconda3\lib
\site-packages (2.26.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\programdata\anaconda
3\lib\site-packages (from requests==2.26.0) (1.26.11)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\sit
e-packages (from requests==2.26.0) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\l
ib\site-packages (from requests==2.26.0) (2022.9.14)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\programdata\anac
onda3\lib\site-packages (from requests==2.26.0) (2.0.4)
Requirement already satisfied: plotly==5.3.1 in c:\programdata\anaconda3\lib\si
te-packages (5.3.1)
Requirement already satisfied: tenacity>=6.2.0 in c:\programdata\anaconda3\lib
\site-packages (from plotly==5.3.1) (8.0.1)
Requirement already satisfied: six in c:\programdata\anaconda3\lib\site-package
s (from plotly==5.3.1) (1.16.0)
```

In [7]:
```
import yfinance as yf
import pandas as pd
import requests
from bs4 import BeautifulSoup
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

In [11]:
```
tesla = yf.Ticker("TSLA")
```

In [13]:
```
tesla_data = tesla.history(period="max")
```

In [12]:
```
tesla_data.reset_index(inplace = True)
tesla_data.head()
```

Out[12]:

	index	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	0	2010-06-29	1.266667	1.666667	1.169333	1.592667	281494500	0	0.0
1	1	2010-06-30	1.719333	2.028000	1.553333	1.588667	257806500	0	0.0
2	2	2010-07-01	1.666667	1.728000	1.351333	1.464000	123282000	0	0.0
3	3	2010-07-02	1.533333	1.540000	1.247333	1.280000	77097000	0	0.0
4	4	2010-07-06	1.333333	1.333333	1.055333	1.074000	103003500	0	0.0

In [14]:
```
url = "https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue"
html_data = requests.get(url).text
```

In [15]:
```

soup = BeautifulSoup(html_data, "html.parser")
soup.find_all('title')
```

Out[15]:
```
[<title>Tesla Revenue 2010-2022 | TSLA | MacroTrends</title>]
```

In [16]:
```
tesla_revenue = pd.DataFrame(columns = ['Date', 'Revenue'])

for row in soup.find_all("tbody")[1].find_all("tr"):
    col = row.find_all("td")
    date = col[0].text
    revenue = col[1].text.replace("$", "").replace(",", "")

    tesla_revenue = tesla_revenue.append({"Date": date, "Revenue": revenue}, ign
```

In [17]:
```
tesla_revenue.dropna(inplace=True)
tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""]
```

In [18]:

```
tesla_revenue.tail()
```

Out[18]:

	Date	Revenue
48	2010-09-30	31
49	2010-06-30	28
50	2010-03-31	21
52	2009-09-30	46
53	2009-06-30	27

In [19]:

```
GameStop = yf.Ticker("GME")
```

In [20]:

```
gme_data = GameStop.history(period = 'max')
```

In [21]:

```
gme_data.reset_index(inplace = True)
gme_data.head()
```

Out[21]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-02-13	1.620128	1.693350	1.603296	1.691666	76216000	0.0	0.0
1	2002-02-14	1.712707	1.716073	1.670625	1.683250	11021600	0.0	0.0
2	2002-02-15	1.683250	1.687458	1.658002	1.674834	8389600	0.0	0.0
3	2002-02-19	1.666418	1.666418	1.578047	1.607504	7410400	0.0	0.0
4	2002-02-20	1.615921	1.662210	1.603296	1.662210	6892800	0.0	0.0

In [22]:

```
url = "https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue"
html_data = requests.get(url).text
```

In [23]:

```
soup = BeautifulSoup(html_data, "html.parser")
soup.find_all('title')
```

Out[23]:

```
[<title>GameStop Revenue 2010-2022 | GME | MacroTrends</title>]
```

In [24]:

```
gme_revenue = pd.DataFrame(columns = ['Date', 'Revenue'])

for row in soup.find_all("tbody")[1].find_all("tr"):
    col = row.find_all("td")
    date = col[0].text
    revenue = col[1].text.replace("$", "").replace(",", "")

    gme_revenue = gme_revenue.append({"Date": date, "Revenue": revenue}, ignore_i
```

In [26]:

```python
gme_revenue.dropna(inplace=True)
gme_revenue = gme_revenue[gme_revenue['Revenue'] != ""]
gme_revenue.tail()
```

Out[26]:

	Date	Revenue
48	2010-09-30	31
49	2010-06-30	28
50	2010-03-31	21
52	2009-09-30	46
53	2009-06-30	27

In [27]:

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_11908\851608974.py in <module>
----> 1 make_graph(tesla_data, tesla_revenue, 'Tesla')

NameError: name 'make_graph' is not defined
```

In [30]:

```python
def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True, subplot_titles=("Histo
    fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data.Date, infer_datetime_for
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data.Date, infer_datetime_f
    fig.update_xaxes(title_text="Date", row=1, col=1)
    fig.update_xaxes(title_text="Date", row=2, col=1)
    fig.update_yaxes(title_text="Price ($US)", row=1, col=1)
    fig.update_yaxes(title_text="Revenue ($US Millions)", row=2, col=1)
    fig.update_layout(showlegend=False,
    height=900,
    title=stock,
    xaxis_rangeslider_visible=True)
    fig.show()
```

In [34]:

```
def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True, subplot_titles=("Histo
    fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data.Date, infer_datetime_for
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data.Date, infer_datetime_f
    fig.update_xaxes(title_text="Date", row=1, col=1)
    fig.update_xaxes(title_text="Date", row=2, col=1)
    fig.update_yaxes(title_text="Price ($US)", row=1, col=1)
    fig.update_yaxes(title_text="Revenue ($US Millions)", row=2, col=1)
    fig.update_layout(showlegend=False,
    height=900,
    title=stock,
    xaxis_rangeslider_visible=True)
    fig.show()
```

In [42]:

```
#installing libraries
!pip install yfinance
!pip install bs4
!pip install lxml
```

```
Requirement already satisfied: yfinance in c:\programdata\anaconda3\lib\site-pa
ckages (0.1.67)
Requirement already satisfied: lxml>=4.5.1 in c:\programdata\anaconda3\lib\site
-packages (from yfinance) (4.9.1)
Requirement already satisfied: multitasking>=0.0.7 in c:\programdata\anaconda3
\lib\site-packages (from yfinance) (0.0.11)
Requirement already satisfied: requests>=2.20 in c:\programdata\anaconda3\lib\s
ite-packages (from yfinance) (2.26.0)
Requirement already satisfied: pandas>=0.24 in c:\programdata\anaconda3\lib\sit
e-packages (from yfinance) (1.3.3)
Requirement already satisfied: numpy>=1.15 in c:\programdata\anaconda3\lib\site
-packages (from yfinance) (1.21.5)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\programdata\anacond
a3\lib\site-packages (from pandas>=0.24->yfinance) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in c:\programdata\anaconda3\lib\sit
e-packages (from pandas>=0.24->yfinance) (2022.1)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\l
ib\site-packages (from requests>=2.20->yfinance) (2022.9.14)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\sit
e-packages (from requests>=2.20->yfinance) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\programdata\anac
onda3\lib\site-packages (from requests>=2.20->yfinance) (2.0.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\programdata\anaconda
3\lib\site-packages (from requests>=2.20->yfinance) (1.26.11)
Requirement already satisfied: six>=1.5 in c:\programdata\anaconda3\lib\site-pa
ckages (from python-dateutil>=2.7.3->pandas>=0.24->yfinance) (1.16.0)
Requirement already satisfied: bs4 in c:\programdata\anaconda3\lib\site-package
s (0.0.1)
Requirement already satisfied: beautifulsoup4 in c:\programdata\anaconda3\lib\s
ite-packages (from bs4) (4.11.1)
Requirement already satisfied: soupsieve>1.2 in c:\programdata\anaconda3\lib\si
te-packages (from beautifulsoup4->bs4) (2.3.1)
Requirement already satisfied: lxml in c:\programdata\anaconda3\lib\site-packag
es (4.9.1)
```

In [37]:

```
#importing libraries
import pandas as pd

import yfinance as yf
import requests
from bs4 import BeautifulSoup

import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

In [38]:

```
def plot_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True, subplot_titles=("Histo
    fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data.Date, infer_datetime_for
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data.Date, infer_datetime_f
    fig.update_xaxes(title_text="Date", row=1, col=1)
    fig.update_xaxes(title_text="Date", row=2, col=1)
    fig.update_yaxes(title_text="Price ($)", row=1, col=1)
    fig.update_yaxes(title_text="Revenue ($ Millions)", row=2, col=1)
    fig.update_layout(showlegend=False, height=1000, title=stock, xaxis_rangeslid
    fig.show()
```

In [39]:

```
# Using the Ticker function to create a ticker object.
# ticker symbol of tesla is TSLA
tesla_data = yf.Ticker('TSLA')

# history function helps to extract stock information.
# setting period parameter to max to get information for the maximum amount of ti
tsla_data = tesla_data.history(period='max')

# Resetting the index
tsla_data.reset_index(inplace=True)

# display the first five rows
tsla_data.head()
```

Out[39]:

```
         Date      Open      High       Low     Close     Volume  Dividends  Stock Splits
0  2010-06-29  1.266667  1.666667  1.169333  1.592667  281494500          0           0.0
1  2010-06-30  1.719333  2.028000  1.553333  1.588667  257806500          0           0.0
2  2010-07-01  1.666667  1.728000  1.351333  1.464000  123282000          0           0.0
3  2010-07-02  1.533333  1.540000  1.247333  1.280000   77097000          0           0.0
4  2010-07-06  1.333333  1.333333  1.055333  1.074000  103003500          0           0.0
```

In [44]:

```
# using requests library to download the webpage
url='https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue'

# Save the text of the response
html_text = requests.get(url).text

# Parse the html data using beautiful_soup.
soup = BeautifulSoup(html_text, "lxml")
```

In [45]:

```
# Using beautiful soup extract the table with Tesla Quarterly Revenue.
# creating new dataframe
tsla_revenue = pd.DataFrame(columns=["Date", "Revenue"])

tables = soup.find_all('table')
table_index=0

for index, table in enumerate(tables):
    if ('Tesla Quarterly Revenue'in str(table)):
        table_index=index

for row in tables[table_index].tbody.find_all("tr"):
    col = row.find_all("td")
    if (col!=[]):
        date =col[0].text
        # to remove comma and dollar sign
        revenue =col[1].text.replace("$", "").replace(",", "")
        tsla_revenue=tsla_revenue.append({'Date':date,'Revenue':revenue},
                                          ignore_index=True)

# displaying dataframe
tsla_revenue
```

Out[45]:

	Date	Revenue
0	2022-09-30	21454
1	2022-06-30	16934
2	2022-03-31	18756
3	2021-12-31	17719
4	2021-09-30	13757
5	2021-06-30	11958
6	2021-03-31	10389
7	2020-12-31	10744
8	2020-09-30	8771
9	2020-06-30	6036
10	2020-03-31	5985
11	2019-12-31	7384
12	2019-09-30	6303
13	2019-06-30	6350
14	2019-03-31	4541
15	2018-12-31	7226
16	2018-09-30	6824
17	2018-06-30	4002
18	2018-03-31	3409
19	2017-12-31	3288
20	2017-09-30	2985

	Date	Revenue
21	2017-06-30	2790
22	2017-03-31	2696
23	2016-12-31	2285
24	2016-09-30	2298
25	2016-06-30	1270
26	2016-03-31	1147
27	2015-12-31	1214
28	2015-09-30	937
29	2015-06-30	955
30	2015-03-31	940
31	2014-12-31	957
32	2014-09-30	852
33	2014-06-30	769
34	2014-03-31	621
35	2013-12-31	615
36	2013-09-30	431
37	2013-06-30	405
38	2013-03-31	562
39	2012-12-31	306
40	2012-09-30	50
41	2012-06-30	27
42	2012-03-31	30
43	2011-12-31	39
44	2011-09-30	58
45	2011-06-30	58
46	2011-03-31	49
47	2010-12-31	36
48	2010-09-30	31
49	2010-06-30	28
50	2010-03-31	21
51	2009-12-31	
52	2009-09-30	46
53	2009-06-30	27

In [46]:

```
# removing null values
tsla_revenue = tsla_revenue[tsla_revenue['Revenue']!='']
tsla_revenue
```

Out[46]:

	Date	Revenue
0	2022-09-30	21454
1	2022-06-30	16934
2	2022-03-31	18756
3	2021-12-31	17719
4	2021-09-30	13757
5	2021-06-30	11958
6	2021-03-31	10389
7	2020-12-31	10744
8	2020-09-30	8771
9	2020-06-30	6036
10	2020-03-31	5985
11	2019-12-31	7384
12	2019-09-30	6303
13	2019-06-30	6350
14	2019-03-31	4541
15	2018-12-31	7226
16	2018-09-30	6824
17	2018-06-30	4002
18	2018-03-31	3409
19	2017-12-31	3288
20	2017-09-30	2985
21	2017-06-30	2790
22	2017-03-31	2696
23	2016-12-31	2285
24	2016-09-30	2298
25	2016-06-30	1270
26	2016-03-31	1147
27	2015-12-31	1214
28	2015-09-30	937
29	2015-06-30	955
30	2015-03-31	940
31	2014-12-31	957
32	2014-09-30	852

	Date	Revenue
33	2014-06-30	769
34	2014-03-31	621
35	2013-12-31	615
36	2013-09-30	431
37	2013-06-30	405
38	2013-03-31	562
39	2012-12-31	306
40	2012-09-30	50
41	2012-06-30	27
42	2012-03-31	30
43	2011-12-31	39
44	2011-09-30	58
45	2011-06-30	58
46	2011-03-31	49
47	2010-12-31	36
48	2010-09-30	31
49	2010-06-30	28
50	2010-03-31	21
52	2009-09-30	46
53	2009-06-30	27

In [47]:

```
plot_graph(tsla_data, tsla_revenue, 'Tesla Historical Share Price & Revenue')
```

In [48]:

```
#  ticker symbol of GameStop is GME
gamestop = yf.Ticker('GME')

# extracting stock information
gme_data=gamestop.history(period='max')

#reset the index
gme_data.reset_index(inplace=True)
gme_data.head()
```

Out[48]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-02-13	1.620129	1.693350	1.603296	1.691667	76216000	0.0	0.0
1	2002-02-14	1.712707	1.716074	1.670626	1.683250	11021600	0.0	0.0
2	2002-02-15	1.683250	1.687458	1.658001	1.674834	8389600	0.0	0.0
3	2002-02-19	1.666418	1.666418	1.578047	1.607504	7410400	0.0	0.0
4	2002-02-20	1.615921	1.662210	1.603296	1.662210	6892800	0.0	0.0

In [50]:

```
# using requests library to download the webpage
url = 'https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue'

# Save the text of the response
html_data = requests.get(url).text

# parse the html data
soup=BeautifulSoup(html_data, 'lxml')
```

In [51]:

```
# Using beautiful soup extract the table with GameStop Quarterly Revenue
# creating new dataframe
gme_revenue = pd.DataFrame(columns=["Date", "Revenue"])
tables = soup.find_all('table')

table_index=0
for index, table in enumerate(tables):
    if ('GameStop Quarterly Revenue'in str(table)):
        table_index=index

for row in tables[table_index].tbody.find_all("tr"):
    col = row.find_all("td")
    if (col!=[]):
        date =col[0].text
        # comma and dollar sign is removed
        revenue =col[1].text.replace("$", "").replace(",", "")
        gme_revenue=gme_revenue.append({'Date':date,'Revenue':revenue},
                                       ignore_index=True)

gme_revenue.head()
```

Out[51]:

```
         Date  Revenue
0  2022-07-31     1136
1  2022-04-30     1378
2  2022-01-31     2254
3  2021-10-31     1297
4  2021-07-31     1183
```

In [52]:

```
plot_graph(gme_data, gme_revenue, 'GameStop')
```

```
In [ ]:
```